

Claims

What I claim as my invention is:

1. A method in a computer system for storing element chunks of elements in a storage area, comprising:
 - storing at least one root storage in a storage area;
 - storing a model directory in said at least one root storage;
 - 5 storing at least one model in said model directory
 - storing a graphic element list and a control element list in said at least one model, said graphic element list having element chunks containing graphic elements and said control element list having element chunks containing control elements;
 - 10 assigning a preselected number of elements to each element chunk; and
 - allocating each element to an element chunk in one of said control element list and said graphic element list.
2. The method of claim 1, further comprising:
 - compressing each element chunk; and
 - storing at least one compressed element chunk in at least one of said graphic element list and said control element list.
3. The method of claim 1, further comprising:
 - encrypting each element chunk; and
 - storing at least one encrypted element chunk in at least one of said graphic element list and said control element list.
4. The method of claim 1, further comprising:
 - compressing and encrypting each element chunk; and
 - storing at least one compressed and encrypted element chunk in at least one of said graphic element list and said control element list.
5. The method of claim 1, wherein said preselected number is a maximum number of elements.

6. The method of claim 1, further comprising the steps of:
creating an additional element chunk when the number of elements
exceeds said preselected number of elements assigned to each element chunk;
assigning said preselected number of elements to said additional element
5 chunk; and
storing new elements in said additional element chunk.

7. The method of claim 6, further comprising:
compressing each new element chunk; and
storing each new compressed element chunk in at least one of said
graphic element list and said control element list.

8. The method of claim 6, further comprising:
encrypting each new element chunk; and
storing each new encrypted element chunk in at least one of said graphic
element list and said control element list.

9. The method of claim 6, further comprising:
compressing and encrypting each new element chunk; and
storing each new compressed and encrypted element chunk in at least
one of said graphic element list and said control element list.

10. The method of claim 6, wherein said preselected number is a maximum
number of elements.

11. The method of claim 1, further comprising the step of associating a header
with said at least one root storage.

12. The method of claim 1, wherein the computer system is the Internet.

13. The method of claim 1, wherein the computer system is an Intranet.

14. The method of claim 1, wherein the computer system is a local area network.

15. The method of claim 1, wherein said storage area is a file.

16. The method of claim 1, wherein said storage area is adapted to be operable with a computer aided design program.

17. The method of claim 1, further comprising storing in said root storage at least one of a first stream containing a header, a second stream containing session information, a third stream containing a manifest and a fourth stream containing file properties.

18. The method of claim 1, further comprising the step of storing in the root storage at least one of a stream and a storage, neither of which are contained in the model directory.

19. A method in a computer system for storing element chunks of elements in a storage area, comprising:

storing at least one root storage in a storage area;

storing at least one model directory in each root storage;

storing a control model in each root storage;

storing at least one model in said model directory;

storing a graphic element list and a control element list in each model and each control model directory, said graphic element list having element chunks containing graphic elements and said control element list having element chunks containing control elements;

assigning a preselected number of elements to each element chunk; and

allocating each element to an element chunk in one of said control element list and said graphic element list.

20. The method of claim 19, further comprising:
compressing each element chunk; and
storing at least one compressed element chunk in at least one of said
graphic element list and said control element list.
21. The method of claim 19, further comprising:
encrypting each element chunk; and
storing at least one encrypted element chunk in at least one of said
graphic element list and said control element list.
22. The method of claim 19, further comprising:
compressing and encrypting each element chunk; and
storing at least one compressed and encrypted element chunk in at least
one of said graphic element list and said control element list.
23. The method of claim 19, wherein said preselected number is a maximum
number of elements.
24. The method of claim 19, further comprising the steps of:
creating an additional element chunk when the number of elements
exceeds said preselected number of elements assigned to each element chunk;
assigning a preselected number of elements to said additional element
chunk; and
storing new elements in said additional element chunk.
25. The method of claim 24, further comprising:
compressing each additional element chunk; and
storing each additional compressed element chunk in at least one of said
graphic element list and said control element list.

26. The method of claim 24, further comprising:
encrypting each additional element chunk; and
storing each additional encrypted element chunk in at least one of said
graphic element list and said control element list.
27. The method of claim 24, further comprising:
compressing and encrypting each additional element chunk; and
storing each additional compressed and encrypted element chunk in at
least one of said graphic element list and said control element list.
28. The method of claim 24, wherein said preselected number is a maximum
number of elements.
29. The method of claim 19, further comprising the step of associating a
header with said at least one root storage.
30. The method of claim 19, wherein the computer system is the Internet.
31. The method of claim 19, wherein the computer system is an Intranet.
32. The method of claim 19, wherein the computer system is a local area
network.
33. The method of claim 19, wherein said storage area is a file.
34. The method of claim 19, wherein said storage area is adapted to be
operable with a computer aided design program.
35. The method of claim 19, further comprising storing in said root storage at
least one of a first stream containing a header, a second stream containing session
information, a third stream containing a manifest and a fourth stream containing file
properties.

36 The method of claim 19, further comprising the step of storing at least one of a stream and a storage, neither of which are contained in the model directory, in the root storage.

37. A method in a computer system for modifying at least one compressed element chunk in a main storage area, comprising:

storing at least one root storage in the main storage area;

storing a model directory in said at least one root storage;

5 storing at least one model including a graphic element list and a control element list in each model directory;

storing at least one compressed element chunk in each graphic element list and control element list, wherein each said compressed element chunk comprises compressed elements;

10 storing said at least one root storage in a temporary memory;

decompressing said compressed element chunks and compressed elements in said graphic element list and said control element list;

modifying elements in the temporary memory;

15 flagging said decompressed element chunks having said modified elements in said temporary memory with a dirty flag;

compressing said flagged element chunks to provide modified compressed element chunks after a predetermined idle time has passed; and

replacing said compressed element chunks in the main storage area with said modified compressed element chunks.

38. The method as in claim 37, wherein the computer system comprises a first workstation having said temporary memory, wherein said first work station accesses and modifies elements; and at least a second workstation accesses and views said modified elements upon request.

39. The method of claim 37, further comprising the step of associating a header with each root storage.

40. The method of claim 37, wherein the computer system is the Internet.
41. The method of claim 37, wherein the computer system is an Intranet.
42. The method of claim 37, wherein the computer system is a local area network.
43. The method of claim 37, wherein said main storage area is a file.
44. The method of claim 37, wherein at least one of said root storages is adapted to be operable with a computer aided design program.
45. The method of claim 37, wherein each said root storage is a file.
46. The method of claim 37, further comprising the step of associating at least one of a stream and a storage, neither of which are contained in the model directory, with at least one of the root storages.
47. The method of claim 37, further comprising the step of storing in said at least one root storage at least one of a first stream containing a header, a second stream containing session information, a third stream containing a manifest and a fourth stream containing file properties.
48. The method of claim 37, further comprising the step of storing in said at least one of root storage a control model containing a control model header, a global control element list and a global graphic element list, wherein said global control element list and said global graphic element list contain element chunks including global elements, wherein said global elements are accessed for use in any model directory.
49. The method of claim 37, further comprising the step of automatically sending a signal from the storage area to other workstations to notify users that an element chunk in a root storage had been modified.

50. A method in a computer system for modifying at least one encrypted element chunk stored in a main storage area, comprising:

storing at least one root storage in the main storage area;

storing a model directory in each root storage;

5 storing at least one model including a graphic element list and a control element list in each model directory;

storing at least one encrypted element chunk in each graphic element list and control element list, wherein each element chunk comprises compressed elements;

storing the at least one root storage in a temporary memory;

10 decrypting said element chunks and said elements in the graphic element list and control element list;

modifying elements in said temporary memory;

flagging said element chunks having said modified elements in said temporary memory with a dirty flag;

15 encrypting said element chunks having said modified elements in temporary memory to provide modified encrypted element chunks after a predetermined idle time has passed; and

replacing said encrypted element chunks in the main storage area with said modified encrypted element chunks.

51. The method of claim 50, wherein the computer system comprises a first workstation having said temporary memory, wherein said first work station accesses and modifies elements; and at least a second workstation accesses and views the modified elements upon request.

52. The method of claim 50, further comprising the step of associating a header with each root storage.

53. The method of claim 50, wherein the computer system is the Internet.

54. The method of claim 50, wherein the computer system is an Intranet.

55. The method of claim 50, wherein the computer system is a local area network.

56. The method of claim 50, wherein the main storage area is a file.

57. The method of claim 50, wherein at least one of the root storages is adapted to be operable with a computer aided design program.

58. The method of claim 50, wherein each root storage is a file.

59. The method of claim 50, further comprising the step of associating at least one of a stream and a storage, neither of which are contained in the model directory, with at least one of the root storages.

60. The method of claim 50, further comprising the step of storing in the at least one root storage at least one of a first stream containing a header, a second stream containing session information, a third stream containing a manifest and a fourth stream containing file properties.

61. The method of claim 50, further comprising the step of storing in the at least one of root storage a control model containing a control model header, a global control element list and a global graphic element list, wherein the global control element list and the global graphic element list contain element chunks including global elements, wherein the global elements are accessed for use in any model directory.

62. The method of claim 50, further comprising the step of automatically sending a signal from the storage area to other workstations to notify users that an element chunk in a root storage had been modified.

63. A method in a computer system for modifying at least one compressed encrypted element chunk in a main storage area, comprising:

storing at least one root storage in the main storage area;

storing a model directory in each root storage;

5 storing at least one model including a graphic element list and a control element list in each model directory;

storing at least one compressed and encrypted element chunk in each graphic element list and control element list, wherein each element chunk comprises compressed elements;

10 storing the at least one root storage in a temporary memory;
decrypting the element chunks and elements in the graphic element list and control element list;

decompressing the element chunks and elements in the graphic element list and control element list;

15 modifying elements in the temporary memory;

flagging the element chunks having the modified elements in the temporary memory with a dirty flag;

20 compressing and encrypting the element chunks having the modified elements in temporary memory to provide modified compressed encrypted element chunks after a predetermined idle time has passed; and

replacing the compressed encrypted element chunks in the main storage area with the modified compressed encrypted element chunks.

64. The method as in claim 63, wherein the computer system comprises a first workstation having the temporary memory, wherein the first work station accesses and modifies elements; and at least a second workstation accesses and views the modified elements upon request.

65. The method of claim 63, further comprising the step of associating a header with each root storage.

66. The method of claim 63, wherein the computer system is the Internet.

67. The method of claim 63, wherein the computer system is an Intranet.

68. The method of claim 63, wherein the computer system is a local area network.

69. The method of claim 63, wherein the main storage area is a file.

70. The method of claim 63, wherein at least one of the root storages is adapted to be operable with a computer aided design program.

71. The method of claim 63, wherein each root storage is a file.

72. The method of claim 63, further comprising the step of associating at least one of a stream and storage, neither of which are contained in the model directory, with at least one of the root storages.

73. The method of claim 63, further comprising the step of storing in the at least one root storage at least one of a first stream containing a header, a second stream containing session information, a third stream containing a manifest and a fourth stream containing file properties.

74. The method of claim 63, further comprising the step of storing in the at least one of root storage a control model containing a control model header, a global control element list and a global graphic element list, wherein the global control element list and the global graphic element list contain element chunks including global elements, wherein the global elements are accessed for use in any model directory.

75. The method of claim 63, further comprising the step of automatically sending a signal from the storage area to other workstations to notify users that an element chunk in a root storage had been modified.

76. A computer system for modifying at least one compressed chunk of elements stored in a main storage area, comprising:

- a temporary memory;
- a root storage stored in the main storage area and the temporary memory;
- a model directory stored in the root storage;
- at least one model stored in the model directory;
- at least one graphic element list stored in each model;
- at least one control element list stored in each model;
- at least one compressed element chunk stored in each graphic element

list and control element list, wherein each element chunk comprises compressed elements;

means for decompressing the element chunks and elements in the graphic element list and control element list in the temporary memory;

means for modifying elements in the temporary memory;

means for lagging the element chunks having the modified elements in the temporary memory;

means for compressing the element chunks having the modified elements in temporary memory to provide modified compressed element chunks after a predetermined idle time has passed; and

means for replacing the compressed element chunks in the main storage area with the modified compressed element chunks.

77. The computer system as in claim 76, further comprising a first workstation having the temporary memory, wherein the first work station modifies elements; and at least a second workstation accesses and views the modified elements upon request.

78. The computer system as in claim 76, further comprising means for associating a header with each root storage.

79. The computer system as in claim 76, wherein the computer system is the Internet.

80. The computer system as in claim 76, wherein the computer system is an Intranet.

81. The computer system as in claim 76, wherein the computer system is a local area network.

82. The computer system as in claim 76, wherein the main storage area is a file.

83. The computer system as in claim 76, wherein at least one of the root storages is adapted to be operable with a computer aided design program.

84. The computer system as in claim 76, wherein each root storage is a file.

85. The computer system as in claim 76, further comprising means for associating at least one of a stream and storage, which are not contained in the model directory, with at least one of the root storages.

86. The computer system as in claim 76, further comprising means for storing in the at least one root storage at least one of a first stream containing a header, a second stream containing session information, a third stream containing a manifest and a fourth stream containing file properties.

87. The computer system as in claim 76, further comprising means for storing in the at least one of root storage a control model containing a control model header, a global control element list and a global graphic element list, wherein the global control element list and the global graphic element list contain element chunks including global elements, wherein the global elements are accessed for use in any model directory.

88. The computer system as in claim 76, further comprising means for automatically sending a signal from the storage area to other workstations to notify users that an element chunk in a root storage had been modified.

89. A computer system for modifying at least one encrypted element chunk stored in a main storage area, comprising:

- a temporary memory;
- a root storage stored in the main storage area and the temporary memory;
- a model directory stored in the root storage;
- at least one model stored in the model directory;
- at least one graphic element list stored in each model;
- at least one control element list stored in each model;
- at least one encrypted element chunk stored in each graphic element list

and control element list, wherein each element chunk comprises encrypted elements;

- means for decrypting the element chunks and elements in the graphic element list and control element list in the temporary memory;
- means for modifying elements in the temporary memory;
- means for flagging the element chunks having the modified elements in the temporary memory with a dirty flag;
- means for encrypting the element chunks having the modified elements in temporary memory to provide modified encrypted element chunks after a predetermined idle time has passed; and
- means for replacing the encrypted element chunks in the main storage area with the modified encrypted element chunks.

90. The computer system as in claim 89, further comprising a first workstation having the temporary memory, wherein the first work station modifies elements; and at least a second workstation accesses and views the modified elements upon request.

91. The computer system as in claim 89 further comprising means for associating a header with each root storage.

92. The computer system as in claim 89, wherein the computer system is the Internet.

93. The computer system as in claim 89, wherein the computer system is an Intranet.

94. The computer system as in claim 89, wherein the computer system is a local area network.

95. The computer system as in claim 89, wherein the main storage area is a file.

96. The computer system as in claim 89, wherein at least one of the root storages is adapted to be operable with a computer aided design program.

97. The computer system as in claim 89, wherein each root storage is a file.

98. The computer system as in claim 89, further comprising means for associating at least one of a stream and storage, which are not contained in the model directory, with at least one of the root storages.

99. The computer system as in claim 89, further comprising means for storing in the at least one root storage at least one of a first stream containing a header, a second stream containing session information, a third stream containing a manifest and a fourth stream containing file properties.

100. The computer system as in claim 89, further comprising means for storing in the at least one of root storage a control model containing a control model header, a global control element list and a global graphic element list, wherein the global control element list and the global graphic element list contain element chunks including global elements, wherein the global elements are accessed for use in any model directory.

101. The computer system as in claim 89, further comprising means for automatically sending a signal from the storage area to other workstations to notify users that an element chunk in a root storage had been modified.

102. A computer system for modifying at least one encrypted compressed element chunk stored in a main storage area, comprising:

- a temporary memory;
- a root storage stored in the main storage area and the temporary memory;
- a model directory stored in the root storage;
- at least one model stored in the model directory;
- at least one graphic element list stored in each model directory;
- at least one control element list stored in each model directory;
- at least one encrypted compressed element chunk stored in each graphic

element list and control element list, wherein each element chunk comprises encrypted compressed elements;

means for decrypting the element chunks and elements in the graphic element list and control element list in the temporary memory;

means for decompressing the element chunks and elements in the graphic element list and control element list in the temporary memory;

means for modifying elements in the temporary memory;

means for flagging the element chunks having the modified elements in the temporary memory with dirty flags;

means for compressing and encrypting the element chunks having the modified elements in temporary memory to provide modified encrypted compressed element chunks after a predetermined idle time has passed; and

means for replacing the encrypted compressed element chunks in the main storage area with the modified encrypted compressed element chunks.

103. The computer system of claim 102, further comprising first workstation having the temporary memory, wherein the first work station accesses and modifies elements; and at least second workstation receiving, decrypting, decompressing and viewing the modified elements upon request.

104. The computer system of claim 102, further comprising means for associating a header with each root storage.

105. The computer system of claim 102, wherein the computer system is the Internet.

106. The computer system of claim 102, wherein the computer system is an Intranet.

107. The computer system of claim 102, wherein the computer system is a local area network.

108. The computer system of claim 102, wherein the main storage area is a file.

109. The computer system of claim 102, wherein at least one of the root storages is adapted to be operable with a computer aided design program.

110. The computer system of claim 102, wherein each root storage is a file.

111. The computer system of claim 102, further comprising means for associating at least one of a stream and storage, which are not contained in the model directory, with at least one of the root storages.

112. The computer system of claim 102, further comprising means for storing in the at least one root storage at least one of a first stream containing a header, a second stream containing session information, a third stream containing a manifest and a fourth stream containing file properties.

113. The computer system of claim 102, further comprising means for storing in the at least one of root storage a control model containing a control model header, a global control element list and a global graphic element list, wherein the global control element list and the global graphic element list contain element chunks including global elements, wherein the global elements are accessed for use in any model directory.

114. The method of claim 102, further comprising the steps of automatically sending a signal from the storage area to other workstations to notify users that an element chunk in a root storage had been modified.

115. A method of reading a file in a computer system, comprising:

creating the file by:

storing at least one root storage in a storage area;

storing a model directory in said at least one root storage;

storing at least one model in said model directory

storing a graphic element list and a control element list in said at least one model, said graphic element list having element chunks containing graphic elements and said control element list having element chunks containing control elements;

assigning a preselected number of elements to each element chunk;

allocating each element to an element chunk in one of said control element list and said graphic element list; and

reading the file.

116. A method in a computer system for reading element chunks of elements in a storage area, comprising:

storing at least one root storage in a storage area;

storing at least one model directory in each root storage;

5 storing a control model in each root storage;

storing at least one model in said model directory;

storing a graphic element list and a control element list in each model and each control model directory, said graphic element list having element chunks containing graphic elements and said control element list having element chunks
10 containing control elements;

assigning a preselected number of elements to each element chunk;

allocating each element to an element chunk in one of said control element list and said graphic element list; and

reading one of said element chunks.

FOUO 4/25/2009

117. A method in a computer system for reading a modified compressed element chunk in a main storage area, comprising:

storing at least one root storage in the main storage area;

storing a model directory in said at least one root storage;

5 storing at least one model including a graphic element list and a control element list in each model directory;

storing at least one compressed element chunk in each graphic element list and control element list, wherein each said compressed element chunk comprises compressed elements;

10 storing said at least one root storage in a temporary memory;
decompressing said compressed element chunks and compressed elements in said graphic element list and said control element list;

modifying elements in the temporary memory;

15 flagging said decompressed element chunks having said modified elements in said temporary memory with a dirty flag;

compressing said flagged element chunks to provide modified compressed element chunks after a predetermined idle time has passed;

replacing said compressed element chunks in the main storage area with said modified compressed element chunks; and

20 reading said modified compressed element chunks.